

LIBRARY OF THE
UNIVERSITY OF ILLINOIS
AT URBANA-CHAMPAIGN

510.84

I l 6 r

no.171-187

cop. **2**





Digitized by the Internet Archive
in 2013

<http://archive.org/details/computerscomputi171brea>

COMPUTERS, COMPUTING AND CAREERS

by

H. C. Brearley

March 10, 1965

Revised March 23, 1965

THE LIBRARY OF THE

UNIVERSITY OF ILLINOIS

Urbana, Illinois



OF ILLINOIS · URBANA, ILLINOIS

This volume is bound without no. 177-185

which is/are unavailable.

COMPUTERS, COMPUTING AND CAREERS

by

H. C. Brearley

March 10, 1965

Revised March 23, 1965

THE LIBRARY OF THE

UNIVERSITY OF ILLINOIS

LIBRARY OF THE



DEPARTMENT OF COMPUTER SCIENCE · UNIVERSITY OF ILLINOIS · URBANA, ILLINOIS

510.84
IL6v
no 171-187
cop. 3

TABLE OF CONTENTS

Computer Mechanisms	1
People Who Build Computers	4
Uses of Computers	5
Computer Programs	7
People Who Write Computer Programs	10
Other Computer People	12
Women in Computing	13
Education for a Career in Computing	13
Suggestions for Further Reading	15

FOREWORD

The Department of Computer Science receives numerous inquiries about computers from high school and college students. These questions cover the history and physical form of computers, the uses of computers and career opportunities for computer people. This paper attempts to answer these questions, at least in part and to indicate other sources of information.

Computer Mechanisms

Men have been attempting to reduce the labor of arithmetic calculations for centuries. Some early efforts were directed toward paper-and-pencil techniques such as the development of logarithms in 1614 by James Napier. Other efforts were directed toward simple mechanical devices such as the ancient abacus and the slide rule. Simple adding machines using gears were invented by Blaise Pascal in 1642 and by Gottfried Wilhelm Leibnitz between 1671 and 1694.

Among the first attempts at large-scale mechanized computation were the "difference engine" and the "analytical engine" conceived by the Englishman Charles Babbage (1792-1871). In 1822 Babbage made a small five-digit difference engine which could generate a table of values of any function whose second difference* was a constant, without human intervention except at the start of the problem. He worked, with the support of the British government, on a larger version capable of calculating to 20 places any function whose sixth difference was a constant, but it was never completed. One difficulty was that it required a large number of small precision mechanical parts similar to the gears, ratchets and levers in present-day desk calculators. Before the project was abandoned, much time and money were spent developing the necessary precision machining techniques which were otherwise not available at that time. One author estimates that if made according to plans, the difference engine would have weighed two tons.

*A function whose second difference is a constant is one whose second derivative is a constant, namely a second-degree polynomial. Many tables can be approximated quite accurately over a limited range by a second degree polynomial, and over longer ranges by higher degree polynomials.

In 1834 Babbage conceived the idea of an "analytical engine" which was a mechanical computer capable of performing an indefinitely long sequence of additions, subtractions, multiplications and divisions in any order under the control of sequences of punched paper cards. Babbage estimated that its speed would be one addition per second and one multiplication per minute. Many detailed drawings were made, but the analytical engine was never built.

Another pioneer was the American Herman Hollerith (1860-1929). He refined the principle of storage and transmission of information by means of punched cards which had been used earlier in Babbage's engines and before that in the loom developed by Jacquard for weaving complicated patterns into decorative fabrics. Hollerith developed card-reading, counting and sorting equipment which was used in the United States census of 1890. His name is preserved in the ubiquitous "Hollerith" punched cards, more often referred to as "IBM" cards.

Babbage's efforts in the 1830's were unsuccessful in part because he attempted to extend mechanical techniques such as those used in the now-familiar mechanical desk calculator to much larger and more complex machines. More successful results were obtained during World War II when electrical techniques were used. The first electrical computers used electrical relays to perform the calculations. They also used punched paper tape and electrically operated typewriters, innovations which had been perfected earlier in connection with the printing telegraph network. A number of relay computers were built at Harvard University and Bell Telephone Laboratories. The Harvard Mark I relay computer, which was built in 1939-1944, was 51 feet long, 8 feet high and could add in 0.3 seconds. It was used largely to compute tables.

The relay computers were followed by vacuum tube computers. The first was ENIAC (for Electronic Numerical Integrator and Calculator) built at the University of Pennsylvania at the end of World War II. A tremendous improvement in speed was obtained because a vacuum tube can switch a current on or off in microseconds (millionths of a second) while a relay requires milliseconds (thousandths of a second). A vacuum tube computer called ILLIAC (I) was built at the University of Illinois in 1949-1952. It contained about 2,700 tubes and could perform 1,400 multiplications or 11,000 additions per second. The first mass-produced computers were built in the early 1950's, and they were vacuum tube computers. These included the well-known UNIVAC and IBM 650 computers.

Computers built since the late 1950's use transistors instead of vacuum tubes to achieve increased reliability and speed. One of the larger and faster of the new computers is ILLIAC II, built at the University of Illinois between 1957 and 1963. It can perform 160,000 multiplications per second or 330,000 additions per second. It contains about 50,000 transistors, 125,000 diodes and numerous other parts, and cost about \$4,000,000. Transistor computers are also commercially produced in a variety of sizes. Costs vary from the low ten thousands of dollars for a desk-size computer to millions of dollars for a room-sized computer.

The next generation of computers will continue to use transistors, but may also use integrated circuits (composed of transistors, diodes, resistors and their connections manufactured all at once), super-conducting materials operating at temperatures near absolute zero, and optical techniques.

People Who Build Computers

People who build computers usually work for a computer manufacturer, although a number of one-of-a-kind computers have been built at various university and research laboratories. There are a dozen or so computer manufacturers in the United States, and many more in Europe, Japan and elsewhere.

One important group of people associated with the construction of a computer are the men who determine the organization of the computer, that is, the functions of the various parts and how they are connected to each other. Such people usually have bachelor's, master's or doctor's degrees in mathematics or electrical engineering.

Another very important group designs the electronic circuits of the computer. Most computers are composed of a few simple circuits (one to a dozen transistors each) which are repeated hundreds or thousands of times and connected in many different ways. The circuit designers decide how many basic kinds of circuits are needed. Then they design them, build samples and test them in the dozens of arrangements in which they will be connected in the finished computer. The circuit designers are typically electrical engineers, assisted by technicians. However, a circuit design group working with newer devices such as thin film, superconducting or optical devices may include physicists, metallurgists and others. All except the technicians are usually college graduates, often with advanced degrees.

After the detailed organization of the computer has been determined, and its circuits have been designed, drawings or tables are made showing the

connections of every transistor and every wire. This takes many months and many people. Then one or more prototype computers is constructed from these drawings, which also takes months. At each of these stages difficulties usually arise, which must be corrected. Eventually the design is considered satisfactory and factory production begins.

Uses of Computers

For many years it has been popular to divide computer calculations into two broad classes--scientific and data processing. The scientific class includes such things as calculating the movements of the planets, ballistics of shells, rockets, and satellites, solution of differential equations representing heat or current flow in nonuniform media, calculations in nuclear physics, etc. The data processing class includes sorting and counting as in the census of 1890, payroll calculations, updating inventories, searching lists for pertinent entries, etc.

Some of the computers of the vacuum tube generation were designed specifically for one class or the other. The "scientific" computers emphasized high-speed arithmetic. The "data processing" computers emphasized high-speed input and output. Today this distinction has become less important. Even a "scientific" computer has to read in its data and instructions and print its output, and both may be voluminous. And "data processing" computers have been called upon to perform more operations which require considerable logical (or computing) capability.

A third, more recent class of computer applications is in so-called "real-time" problems. A "real-time" application is one in which the computer is connected directly to some process, receives its input data from the process, and produces output data to control what the process does next. The delay between computer input and computer output is usually fractions of a second. One such process is the control of an oil refinery. The computer may divide its attention among a number of jobs. It may control the raw material and fuel flow so that a cracking tower operates correctly, a fairly simple job. As a more complicated job, it might also decide each week, on the basis of market prices, costs and other data, how much of each grade of gasoline, kerosene and other products to make to maximize profits.

The distinction between the three classes is more academic than real. For example, airlines' reservations computers keep lists of seats sold and unsold, and keep passenger lists for each flight for some months in the future. They also respond via special leased telephone lines to requests for information from keyboards in ticket offices all over the country, and to requests to sell a seat. A response to an inquiry usually takes only a fraction of a minute. Hence, this example embodies both "data processing" and "real-time" aspects.

The extremely wide range of problems which computers have been used to solve might lead one to expect that there are a large variety of computers manufactured. There are a large number of models of computers available from different manufacturers, but the small ones are all more or less alike and the big ones are all more or less alike. Sometimes special features are needed for

special jobs; for example, the airline reservations computer has to have equipment to enable it to communicate over telephone lines to keyboards at the various ticket offices, and it needs an unusually large memory to store all the reservation lists.

But the big differences in the behavior of computers are determined not so much by the physical equipment as by the instructions fed to them. A sequence of these instructions is called a computer program. These instructions are usually punched into cards or paper tape and then fed into the computer. The computer then runs and produces answers in seconds, minutes or hours depending on the length of the program. By feeding programs to a computer one after the other one may make the computer perform first one job, then another. This steady flow of different jobs may continue 24 hours a day in a large installation, interrupted occasionally to test the computer to make sure it is working correctly or to fix a malfunction.

Computer Programs

A human being tells a computer what to do by writing a program. He does this by writing for the computer a sequence of detailed instructions. The instructions have to be detailed, complete and unambiguous. Such phrases as "do it like I told you on Monday," and "you know what I mean," are generally useless.

A computer program is written in a "language." There are several types of computer languages. The most laborious to write is the language the machine itself understands, called machine language. The instructions in

machine language are very elementary. The following examples show some common machine language instructions and one way they might be abbreviated for punching onto cards to feed into the computer.

Add the number in location X in the memory to the number in the accumulator* and leave the result in the accumulator. (ADD X)

Store the contents of the accumulator in location X in the memory. (STO X)

Fetch the number from location X in the memory and put it in the accumulator. (CLA X)

Using these three instruction types, one may write a program to form the sum

$$S = A + B + C + D$$

and store the result in location S in the memory thus:**

CLA	A
ADD	B
ADD	C
ADD	D
STO	S

Computers usually have several dozen types of instructions, and some of them are considerably more complicated than this simple example indicates.

Machine language is laborious to write because it requires that the calculation be broken down into such minute steps. Other computer languages have been developed which are easier for humans to use. One of the more popular

*The accumulator is the main register of the computer, the place where the number being worked on is stored. It corresponds to the main dial of a desk calculator.

**The instructions shown here are actually written in "assembly language" which is a variation of machine language.

is FORTRAN, which stands for formula translator. For example, instead of writing the sequence of five machine language instructions in the example above, one can write in FORTRAN

$$S = A + B + C + D$$

which is both easier to write and easier to read. The hitch is that the computer does not understand FORTRAN, so a FORTRAN program must be translated into machine language before it can be executed. The computer itself performs this translation under the supervision of a translation program. The translation program must be written in machine language, but this has to be done only once. Other computer languages have been developed which are particularly convenient for arithmetic (ALGOL), for manipulation of strings of letters and numbers (LISP), for guidance of computer-controlled tools (APT), and so on.

Most computers can execute about 10,000 to 1,000,000 instructions per second. A human programmer can write and correct about 100 instructions a day. The ratio of speeds is about 30,000,000 to 1. However, it does not require 30,000,000 programmers working full time to keep one computer busy. One reason is that the computer programs contain loops. A loop is a part of a program which is executed over and over again. For example, a program which processes registration data for 20,000 students at a university who take an average of five courses each will contain at least two loops. One loop will be repeated once per student, or 20,000 times. The other loop will be repeated once per course or 100,000 times.

Another reason it does not take 30,000,000 programmers to keep one computer busy is that whole programs are used over and over. The university registration program would be used at the beginning of every semester. A program which processes incoming checks in a bank would be used for several hours each banking day. Also the translation programs are used often, as are many others.

People Who Write Computer Programs

One of the most geographically dispersed occupations associated with computers is computer programming. Programmers work for computer manufacturers and for organizations that rent or buy computers. There is probably at least one computer and one computer programmer in most towns in the United States over 50,000 people.

Activities of programmers vary over a wide range. At one end of the scale is the programmer associated with a small machine at a small factory or bank. He would write most of his programs in one or two languages developed for that machine and that kind of business. The necessary language translation programs will have been supplied by others, frequently by the computer manufacturer. At the other end of the scale the programmer associated with a large installation or a computer manufacturer will write in more languages for more elaborate computers, and he may develop new computer languages and translators for them.

The programmer has the task of translating vague and sometimes contradictory directions from human beings into precise computer instructions

that will lead to the desired result. This is not always easy. It may require great perseverance and tact to get a complete and unambiguous description of the problem from the humans. It also requires diligent attention to detail to write an error-free computer program.

Training in computer programming is available in a variety of forms. Computer manufacturers offer programming courses for their computers, particularly to personnel of organizations that lease or buy them. In some cities "commerical schools" teach programming as they teach desk calculator operation, typing and shorthand. Programming training is available at most colleges and universities, where it varies from four or six-week noncredit "short courses" to one-year credit courses meeting three or four hours a week. Some universities also offer courses in theory of languages, writing translator programs, etc. A few universities offer degree programs in computer science.

A college degree is very desirable for a programmer, but it need not necessarily be in mathematics or computer science. A degree in the field of the prospective employer's business may be just as good. For example, an electrical manufacturing company would probably hire engineers, business administrators and mathematicians as programmers, and an accounting firm would hire accountants and mathematicians. Of course, people who are trained in both computing and in the area of the prospective employer's business are in great demand.

Other Computer People

There are a number of other occupations associated with computers. One of them is servicing computers. Although the individual elements which make up a computer (transistors, resistors, wire, etc.) are very reliable, there are many thousands of them, and every so often one of them fails. Most medium- and large-scale computer installations have their own repairmen or they have a service contract with the manufacturer which assures them of the services of a "field engineer" or "customer engineer" on a few hours notice. The same serviceman often comes once a day or once a week to run programs to check the operation of the computer and to do preventive maintenance. Computer servicemen are usually trained as electronics technicians in trade schools, the armed services, colleges or training courses run by their employer. They live all over the country to be near the computers they service--there is probably at least one living in every city over 100,000.

Computer operators are another occupation. These people load programs into the machine, start it, remove printed output, and sometimes do minor repairs. This is a less skilled job than the computer programmer or serviceman. For the smaller computers, an operator may learn his job from another operator, but for the larger computers, an operator usually attends a manufacturer's training course. Some computer operators work at night, since most large computers are kept running 16 or 24 hours a day.

Women in Computing

Many women are employed in the computer business. Almost every occupation related to computers requires a lot of thinking and very little muscle, so women are in as much demand as men. A large fraction of computer programmers are women. Women mathematicians engage in computer design and hold many other jobs. Women are less numerous in computer engineering and servicing.

Education for a Career in Computing

The educational requirements of various computer occupations have been mentioned in earlier sections. Most of them require a college degree.

A high school student who is headed for a career in computing should take the courses needed for admission to the college and curriculum of his choice. He should study these admission requirements early in his high school career to make sure that he will satisfy them. If after fulfilling these requirements he still has time for more courses, we suggest that he take more college preparatory mathematics courses, and in the case of prospective engineers, more physics and chemistry.

Some college students who are interested in computers enroll in a curriculum in computer science and follow that curriculum. Others enroll in other curricula and take computer-related electives. Those students concerned with computer programming and programming languages should select such electives as programming (frequently offered at two or more levels), theory of compilers, Boolean algebra, set theory, logic, calculus, differential equations,

numerical methods, probability and statistics, linguistics, information storage and retrieval, pattern recognition, etc. Those concerned with computer organization and engineering should select from the courses listed above plus such courses as methods of computer arithmetic, theory of digital machines, automata theory, pulse and logical circuits, solid-state electronics, etc.

At the University of Illinois, computer-related courses are offered by the Department of Computer Science. Administratively this department is a part of the Graduate College, but the contacts with the College of Engineering are very close. For example, essentially all engineering undergraduates take a three-hour course in computers and programming called Introduction to Automatic Digital Computing. The department offers many other courses in computing:

- Introduction to Theory of Digital Machines
- Digital Computer Methods for Data Processing
- Introduction to Numerical Analysis
- Boolean Algebras With Application to Computer Circuits I
- Boolean Algebras With Application to Computer Circuits II
- Intermediate Programming
- Digital Computer Electronics
- Theory of Computer Design and Circuits: Logic
- Theory of Computer Design and Circuits: Memory
- Logical Design and Automatic Digital Computer Circuits
- Advanced Programming
- Mathematical Theory of Data Processing
- Analytical Problems in Numerical Analysis
- Seminar in Automata Theory
- Threshold Logic
- Seminar in Applied Mathematics

Students in the Engineering College can add a strong computational flavor to their curricula by using some of these courses as technical electives. It is also possible

to obtain a bachelor's degree in Mathematics and Computer Science in the College of Liberal Arts and Sciences. This curriculum has a strong mathematical core plus a substantial number of courses drawn from the above list.

Graduate study in computer science is available at a few universities, including the University of Illinois. The work generally consists of two parts. One part is taking advanced courses in subjects such as those mentioned above; the other part is doing individual research on some particular aspect of one of these subjects, and writing up the findings as a thesis.

Suggestions for Further Reading

Encyclopaedia Britannica. See articles on Babbage, binary numbers, computing machines, logarithms, slide rule, tabulating machines.

Benrey, Ronald. Understanding Digital Computers. John F. Rider Publisher, New York, 1964, 166 pp. \$5.45. An introduction to digital computer construction. Many details such as full adders, half adders, instruction decoders described clearly at an elementary level with diagrams and equations. Bibliography. Recommended.

Bernstein, Jeremy. "Profiles - The Analytical Engine, The New Yorker, Oct. 19, 1963, pp. 58-96; Oct. 26, 1963, pp. 54-108. An excellent layman's introduction to the whole computer field. Includes some simple Fortran programs, and considerable history of computer development.

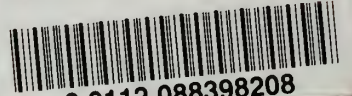
Bowden, B. V. Faster than Thought. Sir Isaac Pitman and Sons, London, 1953, 414 pp. Written by Bowden and a large number of contributors. An extensive survey of British computer technology from the beginning through 1952. A 67-page appendix contains a description of Babbage's analytical engine written in 1842.

Burck, Gilbert. "The Boundless Age of the Computer," Fortune, March 1964, p. 101; April 1964, p. 140; May 1964, p. 153; June 1964, p. 112; Aug. 1964, p. 125; Oct. 1964, p. 120. A series of articles on the economic effects of computers.

- Desmonde, William H. Computers and Their Uses. Prentice-Hall, Englewood Cliffs, N J., 1964, 296 pp. \$10. A survey of the whole field of computing on a fairly elementary level. Over half the book is devoted to verbal descriptions of computer applications. The remainder covers history, machine construction, logic, etc. Unfortunately there is no bibliography.
- Eckert, W. J., and Rebecca Jones. Faster, Faster. McGraw-Hill, New York, 1955, 160 pp. A fairly nontechnical description of the IBM Naval Ordnance Research Calculator (NORC).
- Jacker, Corinne. Man, Memory and Machines. Macmillan, New York, 1964, 126 pp. \$3.95.
- McCracken, D. D. A Guide to Fortran Programming. John Wiley, New York, 1961, 88 pp. \$2.75. A very lucid text on the widely used computer programming language named FORTRAN.
- Morrison, Phillip, and Emily Morrison. Charles Babbage and His Calculating Engines, Selected Writings by Charles Babbage and Others. Dover, New York, 1961, 400 pp. \$2.
- Richards, R. K. Arithmetic Operations in Digital Computers. D. Van Nostrand, New York, 1955. A thorough treatment of computer arithmetic.
- Stibitz, George R., and Jules A. Larrivee. Mathematics and Computers. McGraw-Hill, New York, 1957, 228 pp. \$5. A good, serious introduction to computers, particularly for a reader with some mathematical background. Contains a general bibliography plus a long bibliography on computer applications.



UNIVERSITY OF ILLINOIS-URBANA
510.84 IL6R no. C002 v.171-187(1965)
Illiac III: a processor of visual Inform



3 0112 088398208